



МАГИСТЪРСКИ ФАКУЛТЕТ
ДЕПАРТАМЕНТ "Информатика"

ПРОГРАМА "Софтуерни технологии в Интернет"

КУРС INFM255 Проект по Java за напреднали

**Разработване на игра Судоку
с използване на технологии Java и JavaFX**

Студент:
Емилия Павлова
Фак. №: F46820
Дата: 11.05.2025 г.

Преподавател:
/Тодор Балабанов/

София

Съдържание

- [1. Въведение](#)
- [2. Теоретично изложение](#)
 - [2.1. Същност на проекта](#)
 - [2.2. Алгоритъм за решаване](#)
 - [2.3. Избрани технологии](#)
- [3. Практическа реализация](#)
 - [3.1. Структура на проекта](#)
 - [3.2. Компоненти](#)
- [4. Ръководство за потребителя](#)
 - [4.1. Правила на играта](#)
 - [4.2. Стартиране на приложението](#)
- [5. Заключение](#)
- [6. Информационни източници](#)

1. Въведение

Проектът представлява създаването на графична версия на класическата игра sudoku, използвайки езика за програмиране Java и платформата JavaFX за изграждане на графичния интерфейс на приложението. Изборът на темата се основава на относително ограничения брой функционалности, които са необходими за функциониране на играта, като в същото време предоставя добра възможност за демонстрация приложението на обектно-ориентирано програмиране. Допълнителен бонус е изчистеният и несложен потребителски интерфейс, тъй като той не е основна цел на проекта, а е добавен с цел завършеност на приложението. Включени са функции като генериране на случайни sudoku задачи, проверка за правилни решения, таймер за отчитане на времето, за което е решена задачата.

2. Теоретично изложение

2.1. Същност на проекта

Судоку е логическа игра с числа, която се играе на решетка (грид) с 9 реда и 9 колони, която е разделена на 9 подквадрата (подгридове) 3x3. Цел на играта е да се попълнят всички квадратчета с цифрите от 1 до 9 без да се повтаря никоя в ред, колона или подквадрат. В началото на играта в някои от клетките са подредени определен брой цифри, за да помогнат на играча да попълни правилно пъзела.

Разработеното приложение поддържа следните функционалности:

- Генериране на частично запълнена дъска;
- Разграничаване на фиксирани и въведени от потребителя стойности;

- Валидация при въвеждане на стойност;
- Проверка дали пъзелът е решен правилно;
- Идентифициране на погрешни стойности;
- Възможност за подсказване;
- Възможност за стартиране на нова игра.

2.2. Алгоритъм за решаване

Алгоритъмът за решаване на sudoku обикновено използва подход, наречен "обратно проследяване" (backtracking), при който се опитват различни стойности за всяка клетка. При достигане на противоречие подредбата на дъската се връща стъпка назад и се опитват нови стойности. Този процес продължава до намиране на правилно решение или до изчерпване на всички възможности без да е достигнато такова.

2.3. Избрани технологии

Изграждането на графичен интерфейс за играта Sudoku с език за програмиране Java и платформа JavaFX комбинира обектно-ориентираното програмиране с визуално структуриране на елементи.

Java осигурява стабилна основа за реализиране на логиката на играта. Функционалността е реализирана с помощта на класовете `SudokuBoard`, `SudokuGenerator`, `SudokuCell` и `SudokuSolver`, които отговарят за логическото представяне на дъската, генерирането на валидни начални конфигурации и проверката на решенията.

Свързването между логиката и интерфейса се осъществява чрез събития – следи се за въвеждане на текст или натискане на бутони, които задействат съответните методи за проверка на решение, нова игра или показване на подсказки.

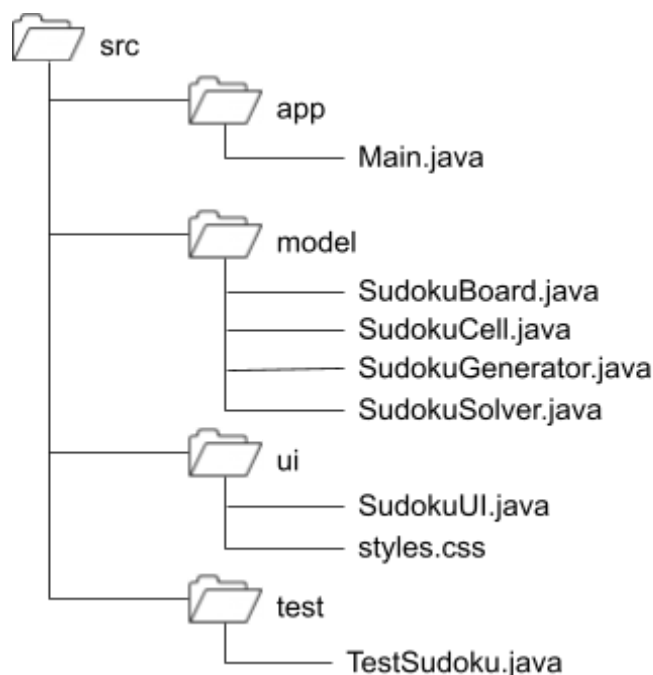
JavaFX предоставя мощна платформа за създаване на интерактивни настолни приложения с богата графика. Чрез JavaFX компоненти се изграждат визуално дъската, отделните ѝ клетки, които позволяват въвеждане на стойности от потребителя, бутоните за контрол над играта, таймерът за отчитане на времето за решение. Чрез стилове, дефинирани в CSS файл, елементите се оформят визуално, за да отличат фиксирани числа, грешки, подсказки, граници на подгридовете и да се придаде приятен за потребителя интерфейс.

3. Практическа реализация

3.1. Структура на проекта

Играта е разработена с помощта на езика за програмиране Java и платформата JavaFX. Архитектурата следва принципите на MVC-модела. Model частта е обособена в класовете `SudokuBoard`, `SudokuCell`, `SudokuGenerator` и `SudokuSolver`, които отговарят за логиката на играта. View-частта е представена чрез JavaFX компонентите в класа

SudokuUI, където се дефинират визуалните елементи и взаимодействията с потребителя. Контролерът не е реализиран като отделен компонент. Неговите функции са интегрирани директно в SudokuUI. Този смесен подход позволява бързо и ефективно разработване на функционалност и е подходящ за по-малки проекти с ограничена сложност като настоящия.



Фиг. 1 Структура на кода на проекта

3.2. Компоненти

Компонентите, които изграждат sudoku-приложението, са вече споменатите класове SudokuCell, SudokuBoard, SudokuSolver, SudokuGenerator, SudokuUI, тестови файл TestSudoku и главният Main файл, който стартира играта.

- SudokuCell - Класът SudokuCell представя модел на отделната клетка на дъската. Той съдържа информация за стойността ѝ, която може да е число от 1 до 9, или 0, ако е празна. Тук се съдържа информация и за това дали клетката е фиксирана - т.е. стойността ѝ е видима още със стартирането на играта и не може да бъде променяна - или е празна и потребителят трябва да намери правилната стойност.
- SudokuBoard - Този клас представя борда на играта. Той управлява цялата 9x9 мрежа от клетки и предоставя методи за достъп до отделните клетки, както и за проверка дали дадено число може да бъде поставено на конкретна позиция без да наруши правилата на сутоду.
- SudokuSolver - Класът SudokuSolver съдържа алгоритъм за решаване на sudoku пъзела. Той използва рекурсивен алгоритъм - backtracking, който опитва

всички възможни стойности за празните клетки, докато намери валидно решение. Процесът, през който минава, е следният:

- Минава се през всички клетки;
 - При празна клетка (стойност 0) се пробват стойности от 1 до 9;
 - Ако стойността е валидна, класът я задава и извиква себе си рекурсивно;
 - Ако в някакъв момент не може да продължи, се връща стъпка назад (backtrack);
 - Ако достигне края без празни клетки, е намерено решение;
 - Ако всички възможни стойности за дадена клетка се изчерпат и няма опция за стъпка назад, решение с текущо въведените стойности не е възможно.
- **SudokuGenerator** - Клас, който генерира (нови) валидни sudoku пъзели. Използва рекурсивен backtracking алгоритъм, за да генерира коректно попълнена sudoku решетка, след което произволно премахва част от стойностите, докато останат зададен брой фиксирани клетки, така че играчът да може да решава пъзела.
- **SudokuUI** - Класът **SudokuUI** реализира графичния интерфейс на играта, като използва JavaFX. Той наследява класа **Application**, което го прави входна точка за стартиране на приложението. **SudokuUI** свързва потребителския интерфейс с логиката на играта. В него се реализират функциите за взаимодействие с потребителя като въвеждане на числа, използване на подсказки, проверка на решение, таймер, стартиране на нова игра.
 При стартиране на приложението в метода `start()` класът генерира нова sudoku дъска с предварително попълнени стойности чрез **SudokuGenerator**, след което създава визуална мрежа от текстови полета (**TextField**) в контейнера от JavaFX **GridPane**. Всяко поле отговаря на клетка от пъзела – ако стойността е предварително зададена, полето се заключва за редакция - това са фиксираните клетки. Фиксираните стойности, подсказките, както грешните стойности при проверка се отличават визуално от останалите стойности чрез стилове. Класът **SudokuUI** поддържа функционалност за проверка на решението - `checkSolution()`, активиране на подсказки - `hintMode` чрез метода `showHint()` и следене на времето с таймер чрез методите `startTimer()` и `resetTimer()`. При проверка на решението се сравняват текущите стойности в полетата със стойностите от решената дъска, генерирана чрез **SudokuSolver**. При успешна проверка се показва съобщение с изминалото време, а при грешно - клетките с несъответствия се маркират в червено чрез стила "error-cell".
- **Main** - Класът **Main** е входната точка на sudoku приложението. Методът `launch()` стартира JavaFX приложението, като използва **SudokuUI** за начален клас на интерфейса.
- **TestSudoku** - Тестовият клас **TestSudoku** съдържа юнит и интеграционни тестове, с които се проверява основната логика на приложението. Тестват се правилното решаване на дъската, валидността на ходовете, правилното клониране на дъската и броят на генерираните числа (подсказки) при създаване на нова игра. Юнит тестовете включват проверки на отделни методи като `isValidMove()` и `clone()`, докато интеграционни тестове като

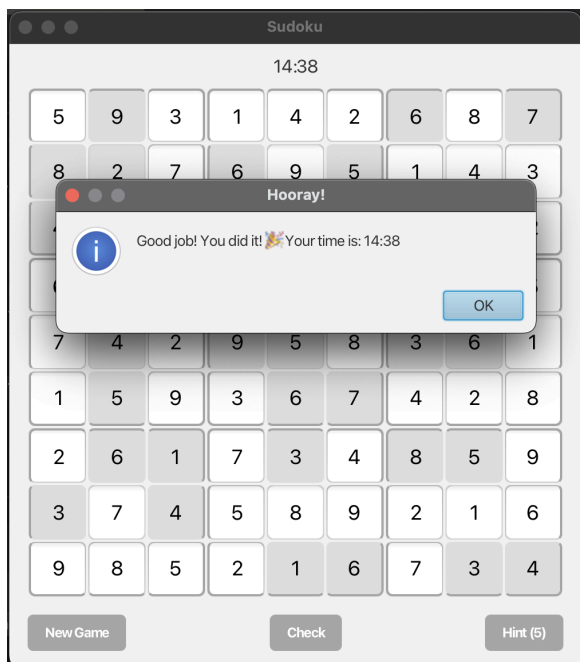
`testSolverOnPresetBoard()` и `testSolveEmptyBoard()` проверяват цялостната функционалност на решаването върху предварително зададена или празна дъска.

4. Ръководство за потребителя

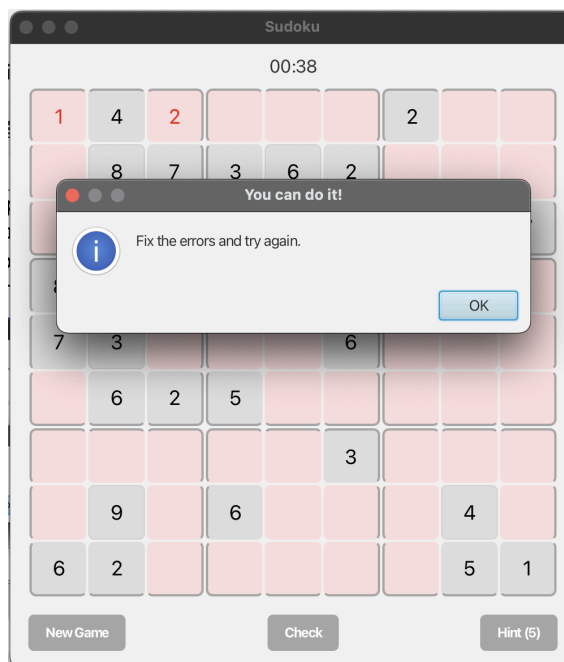
4.1. Правила на играта

Судоку играта представлява дъска с 9 реда и 9 колони, която е разделена на 9 подквадрата с по 3 реда и 3 колони. Във всеки ред, колона и подквадрат трябва да се подредят числата от 1 до 9 без повторения. При стартирането на играта има определен брой попълнени стойности - т.нар. подсказки, които улесняват намирането на решение. Те са фиксирани и не могат да бъдат променяни от потребителя. В текущото приложение броят им всеки път е 30.

Когато потребителят е готов с решение, може да провери неговата валидност с бутона `Check`. Ако решението е валидно, ще се изпише съобщение с поздравление и времето, което е отнело решението. Ако решението не е валидно, всички клетки с грешни или празни стойности се оцветяват в червено.



Фиг. 2 Правилно решен пъзел



Фиг. 3 Пъзел с грешки

При затруднение има възможност да се разкрият стойностите на до 5 клетки. Това става с натискане на бутон `Hint`, който активира “режим подсказки”, след което трябва да се посочи клетката, чиято стойности да се разкрие. След всяка разкрита стойност броят им, изписан на бутона, намалява с един. При изчерпването им бутонът става неактивен.

По всяко време има възможност да се стартира нова игра, което става с натискане на бутона `New Game`. Стартирането ѝ рестартира таймерът и броят подсказки.

4.2. Стартиране на приложението

- Стартиране на приложението:

```
java -jar Sudoku.jar
```

- Стартиране на тестове:

```
java -cp out test.TestSudoku
```

5. Заключение

Разработването с помощта на Java и JavaFX приложение игра sudoku демонстрира познания на принципите на обектно-ориентираното програмиране и графичния интерфейс в Java. Реализираната в сегашния си вид игра включва всички основни функционалности като генериране на нов пъзел, подсказки, проверка на решението, таймер.

Проектът позволява да бъде доразвит в няколко насоки. Възможно е да се създадат нива на сложност, примерно лесно, средно и трудно, които да са обвързани с различен брой фиксирани стойности. Подсказките от бутона също могат да бъдат конфигурируеми.

Бутонът за проверка на решението може да се деактивира до запълване на всички клетки със стойности, при което автоматично да извършва проверка и при правилно решение да се показва време от таймера при запълване на последната клетка, при грешно решение времето продължава да тече без прекъсване. Класация с най-добрите времена също би била добро допълнение. Добра практика от потребителска гледна точка би било и да се добави диалог за потвърждение при стартиране на нова игра, ако текущата не е приключила, за да се избегне неволно нулиране.

6. Информационни източници

1. [Oracle Documentation: Java SE Documentation](#)
2. [JavaFX Documentation](#)
3. [Sudoku Solving Algorithms](#)